QUALITY of SERVICE

Introduction

There are applications (and customers) that demand stronger performance guarantees from the network than ‘‘the best that could be done under the circumstances.’’ Multimedia applications in particular, often need a minimum throughput and maximum latency to work.

- **Quality of service** mechanisms let a network with less capacity meet application requirements.

Issues

Four issues must be addressed to ensure quality of service:

1. What applications **need** from the network.
2. How to **regulate the traffic** that enters the network.
3. How to **reserve resources** at routers to guarantee performance.
4. Whether the network can safely **accept more traffic**.

3

No single technique deals efficiently with all these issues. Instead, a variety of techniques have been developed for use at the network (and transport) layer. Practical quality-of-service solutions combine multiple techniques.

4

Application Requirements

- A stream of packets from a source to a destination is called a **flow** (Clark, 1988). A flow might be all the packets of a connection in a connection-oriented network, or all the packets sent from one process to another process in a connectionless network.
- The needs of each flow can be characterized by four primary parameters: **bandwidth, delay, jitter, and loss.** Together, these determine the **QoS** (**Quality of Service**) the flow requires.

Application Requirements

Application	Bandwidth	Delay	Jitter	Loss
Email	Low	Low	Low	Medium
File sharing	High	Low	Low	Medium
Web access	Medium	Medium	Low	Medium
Remote login	Low	Medium	Medium	Medium
Audio on demand	Low	Low	High	Low
Video on demand	High	Low	High	Low
Telephony	Low	High	High	Low
Videoconferencing	High	High	High	Low

To accommodate a variety of applications, networks may support different categories of QoS. An influential example comes from ATM networks:

1. **Constant bit rate** (e.g., telephony).
2. **Real-time variable bit rate** (e.g., compressed videoconferencing).
3. **Non-real-time variable bit rate** (e.g., watching a movie on demand).
4. **Available bit rate** (e.g., file transfer).

Traffic Shaping

- **Traffic shaping** is a technique for regulating the **average rate and burstiness** of a flow of data that enters the network. The goal is to allow applications to transmit a wide variety of traffic that suits their needs, including some bursts, yet have a simple and useful way to describe the possible traffic patterns to the network.

Traffic Shaping

- When a flow is set up, the user and the network (i.e., the customer and the provider) agree on a certain traffic pattern (i.e., shape) for that flow. Sometimes this agreement is called an **SLA (Service Level Agreement)**, especially when it is made over aggregate flows and long periods of time, such as all of the traffic for a given customer.
- As long as the customer fulfills her part of the bargain and only sends packets according to the agreed-on contract, the provider promises to deliver them all in a timely fashion.
- Packets in excess of the agreed pattern might be dropped by the network, or they might be marked as having lower priority. Monitoring a traffic flow is called **traffic policing**.

9

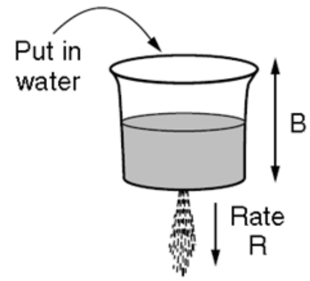
Leaky and Token Buckets

- Now we will look at a general way to characterize traffic, with the leaky bucket and token bucket algorithms. The formulations are slightly different but give an equivalent result.
- Leaky and token buckets limit the long-term rate of a flow but allow short term bursts up to a maximum regulated length to pass through unaltered and without suffering any artificial delays.

10

Leaky Bucket

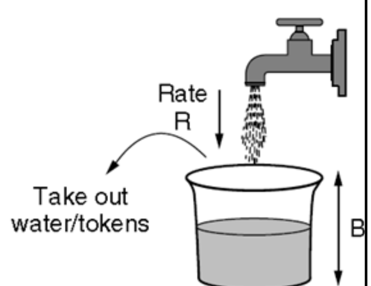
Try to imagine a bucket with a small hole in the bottom. No matter the rate at which water enters the bucket, the outflow is at a constant rate, *R*, when there is any water in the bucket and zero when the bucket is empty. Also, once the bucket is full to capacity *B*, any additional water entering it spills over the sides and is lost.



11

Token Bucket

A different but equivalent formulation is to imagine the network interface as a bucket that is being filled. The tap is running at rate *R* and the bucket has a capacity of *B*, as before. Now, to send a packet we must be able to take water, or tokens, as the contents are commonly called, out of the bucket (rather than putting water into the bucket). No more than a fixed number of tokens, *B*, can accumulate in the bucket, and if the bucket is empty, we must wait until more tokens arrive before we can send another packet.



12

Maximum burst time

Calculating the length of the maximum burst is slightly tricky, because while the burst is being output, more tokens arrive. If we call the burst length *S* sec., the maximum output rate *M* bytes/sec, the token bucket capacity *B* bytes, and the token arrival rate *R* bytes/sec, we can see that an output burst contains a maximum of ***B* + *RS*** byte.

$$B + RS = MS \Rightarrow S = B/(M - R)$$

13

Second Bucket for Smoother Rate

- One way to get smoother traffic is to insert a second token bucket after the first one. The rate of the second bucket should be much higher than the first one. Basically, the first bucket characterizes the traffic, fixing its average rate but allowing some bursts. The second bucket reduces the peak rate at which the bursts are sent into the network.

14

host or Router?

- Using all of these buckets can be a bit tricky. When token buckets are used for traffic shaping at hosts, packets are queued and delayed until the buckets permit them to be sent. When token buckets are used for traffic policing at routers in the network, the algorithm is simulated to make sure that no more packets are sent than permitted.
- Nevertheless, these tools provide ways to shape the network traffic into more manageable forms to assist in meeting quality-of-service requirements.

15

Packet Scheduling

Being able to regulate the shape of the offered traffic is a good start. However, to provide a performance guarantee, we must reserve sufficient resources along the route that the packets take through the network. To do this, we are assuming that the packets of a flow follow the same route.

- As a consequence, **something similar to a virtual circuit has to be set up** from the source to the destination, and all the packets that belong to the flow must follow this route.

16

Resources

Three different kinds of resources can potentially be reserved for different flows:

1. **Bandwidth.** reserving bandwidth means not oversubscribing any output line.
2. **Buffer space.** Up to some maximum value, there will always be a buffer available when the flow needs one.
3. **CPU cycles.** Making sure that the CPU is not overloaded.

17

Queuing Policies

- Each router buffers packets in a queue for each output line until they can be sent, and they are sent in the same order that they arrived. This algorithm is known as **FIFO** (**First-In First-Out**), or equivalently **FCFS** (**First-Come First-Serve**).
- FIFO routers usually drop newly arriving packets when the queue is full. Since the newly arrived packet would have been placed at the end of the queue, this behavior is called **tail drop**.

18

A problem

- FIFO scheduling is simple to implement, but it is not suited to providing good quality of service because when there are multiple flows, one flow can easily affect the performance of the other flows. Many packet scheduling algorithms have been devised that provide stronger isolation between flows

19

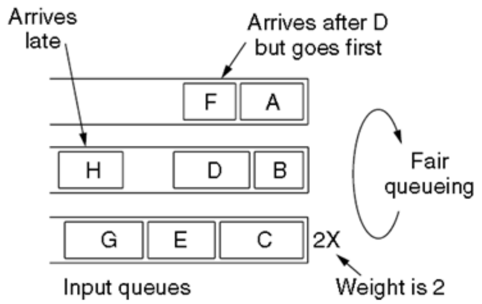
Fair queueing

One of the first ones was the **fair queueing** algorithm devised by Nagle (1987). The essence of this algorithm is that routers have separate queues, one for each flow for a given output line.

20

An example of WFQ

Weighted Fair Queueing



Packet	Arrival time	Length	Finish time	Output order
A	0	8	8	1
B	5	6	11	3
C	5	10	10	2
D	8	9	20	7
E	8	8	14	4
F	10	6	16	5
G	11	10	19	6
H	20	8	28	8

$$F_i = \max(A_i, F_{i-1}) + L_i / W$$

where *Ai* is the arrival time, *Fi* is the finish time, and *Li* is the length of packet *i*.

Admission Control

We have now seen all the necessary elements for QoS and it is time to put them together to actually provide it. QoS guarantees are established through the process of **admission control.**

- The user offers a flow with an accompanying QoS requirement to the network. The network then decides whether to accept or reject the flow based on its capacity and the commitments it has made to other flows. If it accepts, the network reserves capacity in advance at routers to guarantee QoS when traffic is sent on the new flow.

Issues.

Given a path, the decision to accept or reject a flow is not a simple matter of comparing the resources (bandwidth, buffers, cycles) requested by the flow with the router's excess capacity in those three dimensions. It is a little

1. To start with, although some applications may know about their bandwidth requirements, few know about buffers or CPU cycles, so at the minimum, a different way is needed to describe flows and translate this description to router resources. more complicated than that.
2. Next, some applications are far more tolerant of an occasional missed deadline than others. The applications must choose from the type of guarantees that the network can make, whether hard guarantees or behavior that will hold most of the time.
3. Finally, some applications may be willing to haggle about the flow parameters and others may not.

Flow specification

Flows must be described accurately in terms of specific parameters that can be negotiated. A set of such parameters is called a **flow specification**.

- Typically, the sender (e.g., the video server) produces a flow specification proposing the parameters it would like to use. As the specification propagates along the route, each router examines it and modifies the parameters as need be. The modifications can only reduce the flow, not increase it (e.g., a lower data rate, not a higher one).

An example flow specification

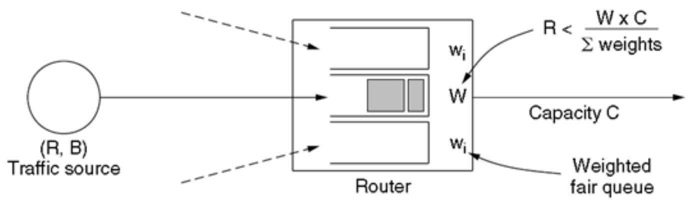
Parameter	Unit
Token bucket rate	Bytes/sec
Token bucket size	Bytes
Peak data rate	Bytes/sec
Minimum packet size	Bytes
Maximum packet size	Bytes

- The minimum size is useful because processing each packet takes some fixed time, no matter how short.
- The maximum packet size is important due to internal network limitations that may not be exceeded.

based on RFCs 2210 and 2211 for Integrated Services

Flow Specification vs. Resources

- One method of relating flow specifications to router resources that correspond to bandwidth and delay performance guarantees is based on traffic sources shaped by (*R*, *B*) token buckets and WFQ at routers. Each flow is given a WFQ weight *W* large enough to drain its token bucket rate *R*. For example, if the flow has a rate of 1 Mbps and the router and output link have a capacity of 1 Gbps, the weight for the flow must be greater than 1/1000th of the total of the weights for all of the flows at that router for the output link.
- if the traffic is saved up in bursts, then a maximum-size burst, *B*, may arrive at the router all at once. In this case the maximum queueing delay, *D*, will be the time taken to drain this burst at the guaranteed bandwidth, or *B*/*R* (again, ignoring packetization effects). If this delay is too large, the flow must request more bandwidth from the network.



27

Integrated Services

- Between 1995 and 1997, IETF put a lot of effort into devising an architecture for streaming multimedia. This work resulted in over two dozen RFCs, starting with RFCs 2205–2212. The generic name for this work is **integrated services**.
- The main part of the integrated services architecture that is visible to the users of the network is **RSVP** **(**The Resource reSerVation Protocol**)**. It is described in RFCs 2205–2210. This protocol is used for making the reservations; other protocols are used for sending the data.

28

IETF Integrated Services

- architecture for providing QOS guarantees in IP networks for individual application sessions
- resource reservation: routers maintain state info of allocated resources,
- admit/deny new call setup requests:

Computer Networking: A Top Down Approach 4th edition.
Jim Kurose, Keith Ross, Addison-Wesley, July 2007.

29

Call Admission

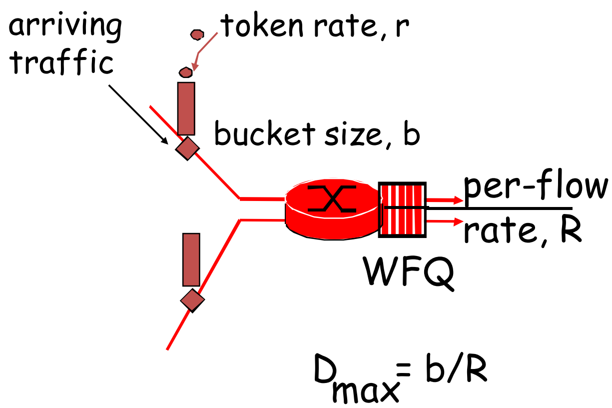
Arriving session must :

- declare its QOS requirement
  - R-spec: defines the QOS being requested,

  R-spec specify what requirements there are for the flow: it can be normal internet 'best effort', in which case no reservation is needed. The 'Controlled Load' setting mirrors the performance of a lightly loaded network: This setting is likely to be used by soft QoS applications. The 'Guaranteed' setting gives an absolutely bounded service, where the delay is promised to never go above a desired amount, and packets never dropped, provided the traffic stays within spec.
- characterize traffic it will send into network
  - T-spec: defines traffic characteristics

  T-spec typically just specify the token rate and the bucket depth.
- signaling protocol: needed to carry R-spec and T-spec to routers (where reservation is required)
  - RSVP

30



RSVP: does not…

- specify how resources are to be reserved
  - rather: a mechanism for communicating needs
- determine routes packets will take
  - that's the job of routing protocols
  - signaling decoupled from routing
- interact with forwarding of packets
  - separation of control (signaling) and data (forwarding) planes

32

RSVP: overview of operation

- sender-to-network signaling
  - *path message:* make sender presence known to routers
  - path teardown: delete sender's path state from routers
- receiver-to-network signaling
  - *reservation message:* reserve resources from sender(s) to receiver
  - reservation teardown: remove receiver reservations
- network-to-end-system signaling
  - path error
  - reservation error

33

Example

- In many multicast applications, groups can change membership dynamically, for example, as people enter a video conference and then get bored and switch to a soap opera or the croquet channel. Under these conditions, the approach of having the senders reserve bandwidth in advance does not work well, since it would require each sender to track all entries and exits of its audience. For a system designed to transmit television with millions of subscribers, it would not work at all.
- RSVP allows multiple senders to transmit to multiple groups of receivers, permits individual receivers to switch channels freely, and optimizes bandwidth use while at the same time eliminating congestion.

34

- In its simplest form, the protocol uses multicast routing using spanning trees, as discussed earlier. Each group is assigned a group address. To send to a group, a sender puts the group's address in its packets. The standard multicast routing algorithm then builds a spanning tree covering all group members. The routing algorithm is not part of RSVP. The only difference from normal multicasting is a little extra information that is multicast to the group periodically to tell the routers along the tree to maintain certain data structures in their memories.

35

RSVP (example)

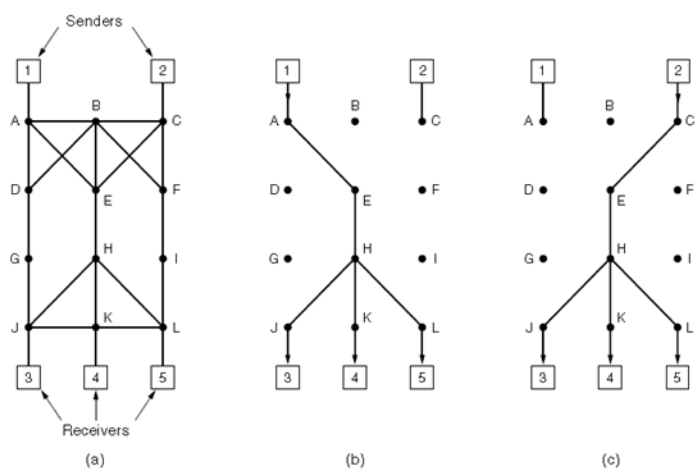


Figure 5-34. (a) A network. (b) The multicast spanning tree for host 1. (c) The multicast spanning tree for host 2.

36

RSVP (example)

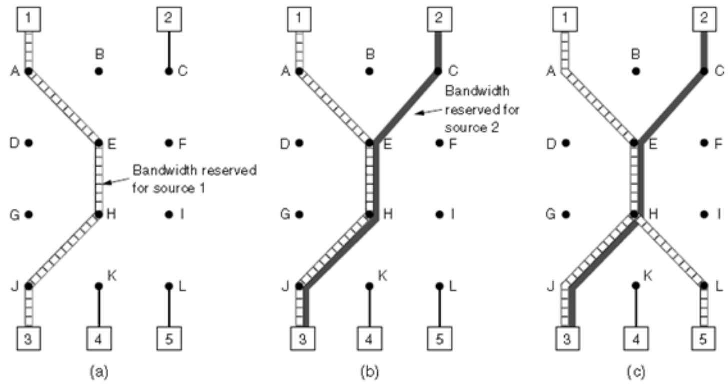


Figure 5-35. (a) Host 3 requests a channel to host 1. (b) Host 3 then requests a second channel, to host 2. (c) Host 5 requests a channel to host 1.

Note that hosts 3 and 5 might have asked for different amounts of bandwidth (e.g., if host 3 is playing on a small screen and only wants the low resolution information), so the capacity reserved must be large enough to satisfy the greediest receiver.

37